DOCKET NO. 99-B-186
CLIENT NO.: STMI01-99186
Customer No. 30425

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of:	Vidyabhusan Gupta
Serial No.:	09/591,621
Filed:	June 9, 2000
For:	SYSTEM AND METHOD FOR DESIGNING AND OPTIMIZING THE MEMORY OF AN EMBEDDED PROCESSING SYSTEM
Technology Center:	2100
Group No.:	2128
Examiner:	Herng Der Day

MAIL STOP APPEAL BRIEF - PATENTS
Commissioner for Patents
P.O. Box 1450
Arlington, VA 22313-1450

Sir:

CERTIFICATE OF MAILING BY FIRST CLASS MAIL

The undersigned hereby certifies that the following documents:

1. Appeal Brief;
2. Fee Transmittal for FY 2006 (in duplicate);
3. Check in the amount of $500.00 (for appeal brief fee); and
4. Postcard receipt

relating to the above application, were deposited as "First Class Mail", with the United States Postal Service, addressed to: MAIL STOP APPEAL BRIEF - PATENTS, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on **November 13, 2006**.

Date: November 13, 2006

Mailer

Date: Nov 13, 2006

William A. Munck, Esq.
Reg. No. 39,308

P.O. Box 802432
Dallas, Texas 75380
Phone: (972) 628-3600
Fax: (972) 628-3616
E-mail: wmunck@munckbutrus.com

PTO/SB/17 (12-04)
Approved for use through 07/31/2006. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

Effective on 12/08/2004.
Fees pursuant to the Consolidated Appropriations Act, 2005 (H.R. 4818).

FEE TRANSMITTAL
For FY 2006

☐ Applicant claims small entity status. See 37 CFR 1.27

TOTAL AMOUNT OF PAYMENT ($) 500.00

Complete if Known	
Application Number	09/591,621
Filing Date	June 9, 2000
First Named Inventor	Vidyabhusan Gupta
Examiner Name	Herng-der Day
Art Unit	2128
Attorney Docket No.	99-B-186 (STMI01-99186)

METHOD OF PAYMENT (check all that apply)

☑ Check ☐ Credit Card ☐ Money Order ☐ None ☐ Other (please identify): _____

☑ Deposit Account Deposit Account Number: 50-0208 Deposit Account Name: Munck Butrus, P.C.

For the above-identified deposit account, the Director is hereby authorized to: (check all that apply)

☐ Charge fee(s) indicated below ☐ Charge fee(s) indicated below, **except for the filing fee**

☑ Charge any additional fee(s) or underpayments of fee(s) under 37 CFR 1.16 and 1.17 ☑ Credit any overpayments

WARNING: Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2038.

FEE CALCULATION

1. BASIC FILING, SEARCH, AND EXAMINATION FEES

	FILING FEES		SEARCH FEES		EXAMINATION FEES		
		Small Entity		Small Entity		Small Entity	
Application Type	Fee ($)	Fee ($)	Fee ($)	Fee ($)	Fee ($)	Fee ($)	Fees Paid ($)
Utility	300	150	500	250	200	100	
Design	200	100	100	50	130	65	
Plant	200	100	300	150	160	80	
Reissue	300	150	500	250	600	300	
Provisional	200	100	0	0	0	0	

2. EXCESS CLAIM FEES

Fee Description	Fee ($)	Small Entity Fee ($)
Each claim over 20 or, for Reissues, each claim over 20 and more than in the original patent	50	25
Each independent claim over 3 or, for Reissues, each independent claim more than in the original patent	200	100
Multiple dependent claims	360	180

Total Claims		Extra Claims	Fee ($)		Fee Paid ($)
_____	- 20 or HP =	_____	x _____	=	_____

HP = highest number of total claims paid for, if greater than 20

Indep. Claims		Extra Claims	Fee ($)		Fee Paid ($)
_____	- 3 or HP =	_____	x _____	=	_____

HP = highest number of independent claims paid for, if greater than 3

Multiple Dependent Claims	
Fee ($)	Fee Paid ($)
_____	_____

3. APPLICATION SIZE FEE

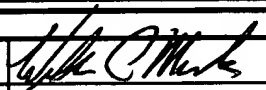
If the specification and drawings exceed 100 sheets of paper, the application size fee due is $250 ($125 for small entity) for each additional 50 sheets or fraction thereof. See 35 U.S.C. 41(a)(1)(G) and 37 CFR 1.16(s).

Total Sheets		Extra Sheets		Number of each additional 50 or fraction thereof		Fee ($)		Fee Paid ($)
_____	- 100 =	_____	/ 50 =	_____	(round **up** to a whole number) x	_____	=	_____

4. OTHER FEE(S)

	Fees Paid ($)
Non-English Specification, $130 fee (no small entity discount)	_____
Other: Appeal Brief fee	$500.00

SUBMITTED BY

Signature		Registration No. (Attorney/Agent) 39,308	Telephone (972) 628-3600
Name (Print/Type)	William A. Munck		Date November 13, 2006

DOCKET NO. 99-B-186
CLIENT NO. STMI01-99186
Customer No. 30425

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of:	Vidyabhusan Gupta
Serial No.:	09/591,621
Filed:	June 9, 2000
For:	SYSTEM AND METHOD FOR DESIGNING AND OPTIMIZING THE MEMORY OF AN EMBEDDED PROCESSING SYSTEM
Technology Center:	2100
Group No.:	2128
Examiner:	Herng-der Day

MAIL STOP APPEAL BRIEF - PATENTS
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

APPEAL BRIEF

The Appellant has appealed to the Board of Patent Appeals and Interferences from the decision of the Examiner dated June 6, 2006, finally rejecting Claims 1-29. The Appellant filed a Notice of Appeal on September 6, 2006, which was received by the U.S. Patent and Trademark Office on September 11, 2006. The Appellant respectfully submits this brief on appeal with the appropriate statutory fee.

TABLE OF CONTENTS

TABLE OF AUTHORITIES

Real Party in Interest

The real party in interest and assignee of this patent application is STMicroelectronics, Inc.

Related Appeals or Interferences

To the best knowledge and belief of the undersigned attorney, there are no appeals or interferences that will directly affect, be directly affected by, or have a bearing on the Board's decision in this pending appeal.

Status of Claims

Claims 1-29 have been finally rejected pursuant to the Office Action dated June 6, 2006. Claims 1-29 are presented on appeal. A copy of all pending claims is provided in Appendix A.

Status of Amendments after Final

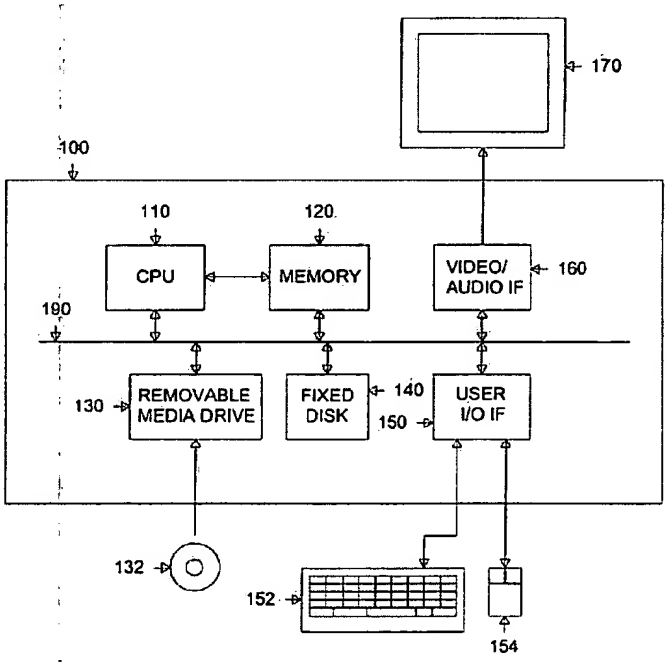
An AMENDMENT AND RESPONSE UNDER 37 C.F.R. § 1.116 was filed on August 4, 2006. According to the Advisory Action dated August 25, 2006, the Examiner entered and considered the AMENDMENT AND RESPONSE.

SUMMARY OF CLAIMED SUBJECT MATTER

The following summary refers to disclosed embodiments and their advantages but does not delimit any of the claimed inventions.

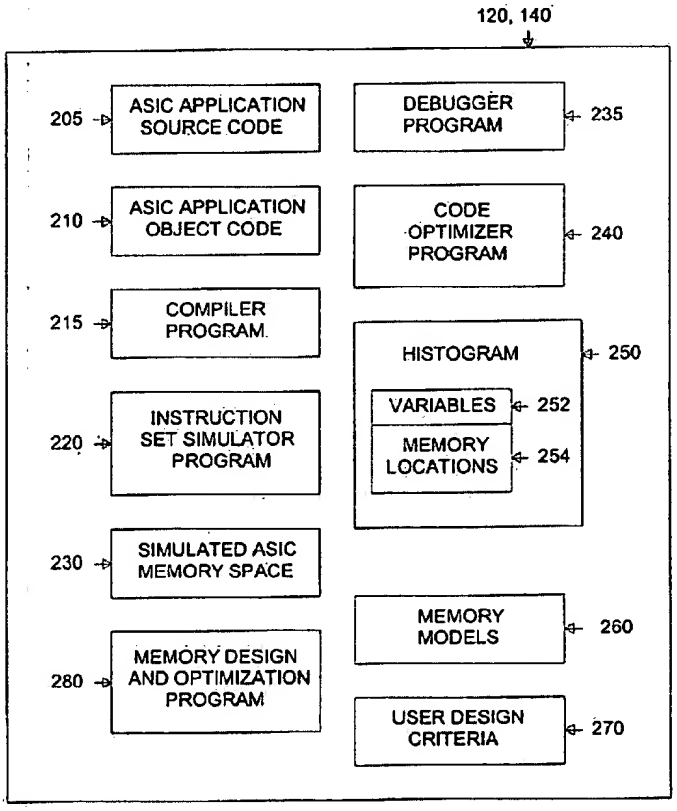
In General

The present application is directed, in general, to a system design tool for determining memory usage of an embedded processing system and for selecting and optimizing the type(s) and size(s) of memory used in the embedded processing system. (*Application, Page 1, Lines 4-8*). An example embodiment of a processing system for determining memory usage and optimizing memory in an embedded processing system is shown in Figure 1 of the Appellant's specification (reproduced below). (*Application, Page 9, Lines 9-17*).



As shown in Figure 1, a processing system 100 represents a computing device having various components. The processing system 100 provides a memory design and optimization apparatus, which could include a controller implemented using a conventional data processor (such as CPU 110) that executes one or more memory design and optimization application programs stored in a memory 120 and a fixed disk drive 140. (*Application, Page 9, Line 18 – Page 11, Line 15*).

Figure 2 of the Appellant's specification (reproduced below) illustrates an example embodiment of the memory design and optimization application programs and data files that may be executed by the processing system 100.



The memory design and optimization application programs are design tools that can be used to determine the optimum memory requirements of a device (such as an ASIC device or other "target device") that is being designed. The processing system 100 simulates the execution of object code to be executed by the target device and determines the types and amounts of memories that may be used in the target device in order to meet user-specified design criteria. The processing system 100 may optionally determine several different memory configurations and may assign to each configuration one or more figures of merit (such as a rating on a scale of 1 to 10) that indicates how well each configuration meets the user-specified design criteria. The processing system 100 may also be capable of modifying the object code in order to achieve an optimum solution of memory devices and software that better meets the user-specified design criteria. (*Application, Page 11, Line 16 – Page 12, Line 15*).

As shown in Figure 2, the memory 120 includes an application source code file 205, an application object code file 210, a compiler program 215, an instruction set simulator (ISS) program 220, a simulated memory space 230, a debugger program 235, a code optimizer program 240, a histogram file 250, a memory models file 260, a user design criteria file 270, and a memory design and optimization program 280. (*Application, Page 12, Line 16 – Page 13, Line 9*).

The application source code file 205 represents the proposed source code written to operate the target device under design. The processing system 100 executes the compiler program 215 to compile the source code and produce executable object code that is stored in the application object code file 210. The ISS program 220 is run using the compiled object code and

simulates the execution of the compiled object code by the target device in the simulated memory space 230. As the execution of the object code is simulated, the debugger 235 works with the ISS program 220 to permit tracking of the simulated execution. (*Application, Page 13, Line 10 – Page 14, Line 4*).

As the execution of the object code is simulated, the ISS program 220 monitors memory access operations and creates histograms of the memory access operations in the histogram file 250. These histograms may include, among other things, a variables histogram file 252 based on variable names contained in the object code and a memory location histogram file 254 based on memory locations accessed by the object code. (*Application, Page 14, Lines 5-14*).

The memory design and optimization program 280 uses the data in the histogram file 250, the data in the memory models file 260, and the data in the user design criteria file 270 to determine the types and amounts of memory that should be used in the target device to best meet the parameters specified by the user in the user design criteria file 270. The data in the user design criteria file 270 could specify general objectives for the target device (such as "minimize SRAM usage," "maximize ROM usage," or "minimize power consumption"). The data in the user design criteria file 270 could also specify more quantitative objectives for the target device (such as a maximum of *N* kilobits of SRAM, a maximum of *R* watts of power consumption, or a maximum write operation access speed). The data in the memory models file 260 specifies the relative performance advantages and disadvantages of different types of memory. (*Application, Page 14, Line 15 – Page 16, Line 8*).

In response to the memory configurations and/or figures of merit determined by the memory design and optimization program 280, the code optimizer 240 may re-order and/or re-write selected portions of the compiled object code in order to achieve greater efficiencies and to better meet the constraints specified in the user design criteria file 270. (*Application, Page 16, Line 9 – Page 17, Line 3*).

Support for Independent Claims

Note that, per 37 C.F.R. § 41.37, only the independent claims are discussed in this section. In the arguments below, however, the dependent claims are also discussed and distinguished from the prior art. The discussion of the claims in this section is for illustrative purposes and is not intended to affect the scope of the claims.

Regarding Claim 1, an apparatus for designing a memory configuration for use in an embedded processing system is provided. The apparatus includes a simulation controller (220) capable of simulating execution of a program to be executed by the embedded processing system. (*Application, Page 13, Line 18 – Page 14, Line 1*). The apparatus also includes a memory access monitor (220) capable of monitoring, during the simulated execution of the program, memory accesses to a simulated memory space (230). (*Application, Page 13, Line 20 – Page 14, Line 14*). The memory access monitor (220) is also capable of generating memory usage statistical data associated with the monitored memory accesses, where the memory accesses include read operations and write operations. (*Application, Page 14, Lines 5-14*). The apparatus further includes a memory optimization controller (280) capable of comparing the

memory usage statistical data and one or more design criteria associated with the embedded processing system and, in response to the comparison, determining at least one memory configuration capable of satisfying the one or more design criteria. (*Application, Page 14, Line 15 – Page 16, Line 8*).

Regarding Claim 8, a method of designing a memory configuration for use in an embedded processing system is provided. The method includes simulating execution of a program to be executed by the embedded processing system. (*Application, Page 19, Lines 3-5*). The method also includes monitoring, during the simulated execution of the program, memory accesses to a simulated memory space, where the memory accesses include read operations and write operations. (*Application, Page 19, Lines 7-9 and 14-21*). The method further includes generating memory usage statistical data associated with the monitored memory accesses. (*Application, Page 19, Lines 7-9 and 14-21*). The method also includes comparing the memory usage statistical data and one or more design criteria associated with the embedded processing system and, in response to the comparison, determining at least one memory configuration capable of satisfying the one or more design criteria. (*Application, Page 19, Line 22 – Page 20, Line 11*).

Regarding Claim 22, a computer-readable storage medium contains computer-executable instructions for designing a memory configuration for use in an embedded processing system. The computer-executable instructions simulate execution of a program to be executed by the embedded processing system. (*Application, Page 19, Lines 3-5*). The computer-executable instructions also monitor, during the simulated execution of the program, memory accesses to a

simulated memory space, where the memory accesses include read operations and write operations. (*Application, Page 19, Lines 7-9 and 14-21*). The computer-executable instructions further generate memory usage statistical data associated with the monitored memory accesses. (*Application, Page 19, Lines 7-9 and 14-21*). In addition, the computer-executable instructions compare the memory usage statistical data and one or more design criteria associated with the embedded processing system and, in response to the comparison, determine at least one memory configuration capable of satisfying the one or more design criteria. (*Application, Page 19, Line 22 – Page 20, Line 11*).

GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

1. Are Claims 1-29 anticipated by Giorgi et al., "An Educational Environment for Program Behavior Analysis and Cache Memory Design" ("*Giorgi*")?

ARGUMENT

Stated Grounds of Rejection

Claims 1-29 stand rejected as being anticipated under 35 U.S.C. § 102(b) by Giorgi et al., "An Educational Environment for Program Behavior Analysis and Cache Memory Design" ("*Giorgi*").

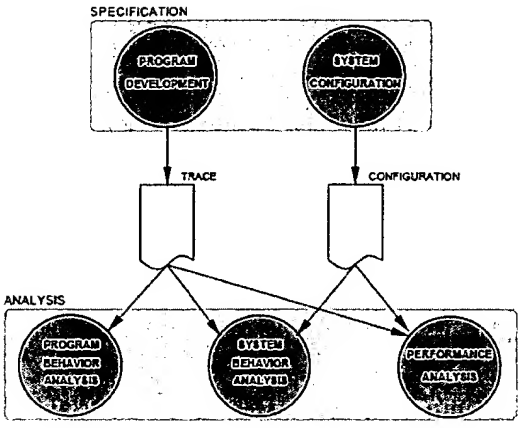
Legal Standards

A prior art reference anticipates a claimed invention under 35 U.S.C. § 102 only if every element of the claimed invention is identically shown in that single reference, arranged as they are in the claims. (*MPEP § 2131; In re Bond*, 910 F.2d 831, 832, 15 U.S.P.Q.2d 1566, 1567 (Fed. Cir. 1990)). Anticipation is only shown where each and every limitation of the claimed invention is found in a single prior art reference. (*MPEP § 2131; In re Donohue*, 766 F.2d 531, 534, 226 U.S.P.Q. 619, 621 (Fed. Cir. 1985)).

Analysis of Examiner's Rejection

The cited reference is briefly discussed in relevant part, and the rejections of the claims are addressed afterwards.

Giorgi recites an educational software package used to analyze the structure and behavior of a cache memory, which helps in the design of cache memories for embedded systems. (*Giorgi, Page 1243, Left column, Abstract*). Figure 1 of *Giorgi* is reproduced below.



The software package supports five different phases as shown in Figure 1. In a "program development" phase, a user builds an application, debugs it, and generates a trace file. In a "system configuration" phase, the user defines the system architecture and the features of each system component (such as selecting a cache memory, a system bus, and memory banks). A "program behavior analysis" phase allows the user to perform trace analyses, such as a trace analysis based on program locality. In a "system behavior analysis" phase, the user carries out a step-by-step simulation by executing one or more memory operations and examining the results. In a "performance analysis" phase, the user can plan and perform a single simulation or an experiment involving varying parameters. (*Giorgi, Page 1244, Left column, Third paragraph – Right column, Second paragraph*).

A program locality analysis (involving both temporal and spatial locality) during the "program behavior analysis" stage could involve the use of 2-D and 3-D graphs. The user can select various parameters for a cache memory to exploit the program locality during the "system configuration" phase, such as cache size and block size. At that point, in the "system behavior

analysis" phase, the user can see how the designed cache memory operates. (*Giorgi, Page 1244, Right column, Last paragraph – Page 1246, Left column, Second paragraph*).

Using the software package, a user can examine how different configurations of cache memory affect program execution. The user can use various graphs to identify the optimal cache memory configuration for given goals (cost-effectiveness and performance requirements). The user can also initiate various simulations to identify other system parameters, such as the cheapest main memory for the selected cache memory configuration. (*Giorgi, Page 1247, Left column, Third paragraph – Right column, Second paragraph*).

Ground of Rejection 1: Claims 1-29 were rejected under 35 U.S.C. § 102(b) as being anticipated by Giorgi et al., "An Educational Environment for Program Behavior Analysis and Cache Memory Design" ("*Giorgi*")

Claim 1

Claim 1 recites an apparatus for designing a memory configuration for use in an embedded processing system, which includes:

> a simulation controller capable of simulating execution of a program to be executed by said embedded processing system;
>
> a memory access monitor capable of monitoring, during said simulated execution of said program, memory accesses to a simulated memory space, wherein said memory access monitor is capable of generating memory usage statistical data associated with said monitored memory accesses, and wherein said memory accesses comprise read operations and write operations; and
>
> a memory optimization controller capable of comparing said memory usage statistical data and one or more design criteria associated with said embedded processing system and, in response to said comparison, determining at least one memory configuration

capable of satisfying said one or more design criteria.

The Examiner fails to show that *Giorgi* anticipates the "memory optimization controller" recited in Claim 1. More specifically, the Examiner fails to show that *Giorgi* anticipates a "memory optimization controller" capable of comparing "memory usage statistical data" and "one or more design criteria" and, in response to the comparison, determining "at least one memory configuration" capable of satisfying the one or more design criteria.

The system of *Giorgi* allows a user to determine if a cache memory is needed and to examine how different cache memory configurations affect the execution of a program. Initially, the user in *Giorgi* must provide various information regarding external memory modules, including configuration parameters such as the module type, the starting address, and the size. Once these parameters are specified, the system of *Giorgi* can perform a simulation, and the user may determine if the simulated results indicate that a cache memory is needed. (*Giorgi, Page 1246, Right column, Third through eighth paragraphs*).

If a cache memory is needed in *Giorgi*, the user must then specify the structure of the cache memory. For example, the user must specify the cache size, block size, number of blocks per set, and replacement policy. The user must also specify the timings of the cache memory and the timings of bus-block operations used by the cache memory. Once the user specifies these parameters, simulations can be performed, and the user can view results and select a configuration. (*Giorgi, Page 1247, Left column, First paragraph – Right column, Second paragraph*).

It is crystal clear here that the system of *Giorgi* does not, in any way, include a "memory optimization controller" capable of determining "at least one memory configuration" in response to comparing "memory usage statistical data" and "one or more design criteria." At best, the system of *Giorgi* simply allows multiple memory configurations to be simulated and allows a user to select a particular memory configuration.

This is inadequate to anticipate Claim 1. The Examiner chose to reject Claim 1 under 35 U.S.C. § 102. § 102 places the burden on the Examiner to show that every single limitation in Claim 1 is disclosed (either inherently or explicitly) in *Giorgi*. The Examiner has not shown that the system of *Giorgi* includes a "memory optimization controller" capable of comparing "memory usage statistical data" and "one or more design criteria" and, in response to the comparison, "determining at least one memory configuration capable of satisfying [the] one or more design criteria" as recited in Claim 1.

For these reasons, *Giorgi* fails to anticipate the Appellant's invention as recited in Claim 1 (and its dependent claims).

Claim 2

Claim 2 recites:

> The apparatus as set forth in Claim 1 wherein said at least one memory configuration is determined from a set of memory types, said set of memory types comprising at least two of static random access memory (SRAM), dynamic random access memory (DRAM), read-only memory (ROM), flash RAM (FLASH), and electronically erasable programmable read-only memory (EEPROM).

Claim 2 depends from Claim 1 and is patentable for the reasons discussed above with respect to Claim 1.

Moreover, the portion of *Giorgi* cited by the Examiner (page 1246, right column, paragraph three) simply refers to various external memory modules in a system, such as a 1-MByte DRAM bank or a 120-KByte PROM bank. *Giorgi* explicitly states that the user selects the external memory modules as part of a system configuration. (*Giorgi, Page 1246, Right column, Third paragraph*). The system of *Giorgi* then performs simulations that allow the user to determine if a cache memory is needed and to select a cache memory configuration. (*Giorgi, Page 1246, Right column, Eighth paragraph – Page 1247, Right column, Second paragraph*).

Giorgi in no way indicates that its system includes a "memory optimization controller" capable of determining "at least one memory configuration" involving SRAM, DRAM, ROM, FLASH, or EEPROM memory types. The selection of the external memory modules (such as the DRAM and PROM memory banks) in *Giorgi* does not appear to involve the use of any simulations or the analysis of simulation results to determine "at least one memory configuration." Rather, the selection simply represents the user's choice of different external memory modules. Not only that, the selection of the external memory modules in *Giorgi* is not performed by a "memory optimization controller," which is capable of "determining at least one memory configuration" in response to comparing "memory usage statistical data" and "one or more design criteria."

As a result, *Giorgi* fails to anticipate a "memory optimization controller" capable of determining, in response to a comparison of memory usage statistical data and one or more

design criteria, "at least one memory configuration" capable of satisfying the one or more design criteria, where the "at least one memory configuration" is determined from a "set of memory types" (including SRAM, DRAM, ROM, FLASH, and EEPROM).

For these reasons, *Giorgi* fails to anticipate the Appellant's invention as recited in Claim 2 (and its dependent claims).

Claim 3

Claim 3 recites:

> The apparatus as set forth in Claim 2 wherein said at least one memory configuration comprises a first memory type and a first memory size associated with said first memory type.

Claim 3 depends from Claim 2 and is patentable for the reasons discussed above with respect to Claim 2.

For these reasons, *Giorgi* fails to anticipate the Appellant's invention as recited in Claim 3 (and its dependent claims).

Claim 4

Claim 4 recites

> The apparatus as set forth in Claim 3 wherein said at least one memory configuration further comprises a second memory type and a second memory size associated with said second memory type.

Claim 4 depends from Claim 3 and is patentable for the reasons discussed above with

respect to Claim 3.

Moreover, Claim 4 (through its dependence from Claim 3) recites that the "at least one memory configuration" includes a "first memory type" and an associated "first memory size" and a "second memory type" and an associated "second memory size." The portion of *Giorgi* cited by the Examiner (page 1246, right column, paragraph seven) simply refers to a user specifying various parameters of the external memory modules (such as the DRAM and PROM modules). These parameters are not based on any type of simulation or any analysis of simulation results. In fact, *Giorgi* indicates that these parameters are selected before any simulations take place. (*See, e.g., Giorgi, Page 1246, Right column, Eighth paragraph*).

The system of *Giorgi* does perform simulations that allow the user to select a cache memory configuration, which is described on page 1247 of *Giorgi*. The simulations, analysis, and other actions related to the selection of the cache memory configuration in *Giorgi* occur after the selection of the external memory modules in *Giorgi*. Also, the Examiner already asserted (with respect to Claim 1) that the "at least one memory configuration" relates to the cache memory configuration selected by the user on page 1247 of *Giorgi*. The selection of the external memory modules on page 1246 of *Giorgi* (which occurs before the simulation and selection of the cache memory configuration) cannot anticipate the determination of the "at least one memory configuration" having multiple memory types and sizes as recited in Claim 4.

For these reasons, *Giorgi* fails to anticipate the Appellant's invention as recited in Claim 4.

Claim 5

Claim 5 recites:

> The apparatus as set forth in Claim 1 wherein said simulation controller simulates execution of said program N times and wherein said memory access monitor monitors said memory accesses during said N simulated executions of said program and generates said memory usage statistical data based on said N simulated executions of said program.

Claim 5 depends from Claim 1 and is patentable for the reasons discussed above with respect to Claim 1.

Moreover, the Examiner asserts that *Giorgi* anticipates these elements of Claim 5 because *Giorgi* recites (i) the generation of a "trace file" (page 1244, left column, paragraph five) and (ii) the cache exits a "cold state" and reaches a "steady condition" (page 1244, right column, paragraph two).

The first cited portion of *Giorgi* simply refers to the analysis of a trace file and does not mention repeated simulation of a program in any way. The second cited portion of *Giorgi* actually recites that the system of *Giorgi* may "simulate an adequate number of memory references without an outcome." This allows a simulated cache memory in the system of *Giorgi* to exit a cold state and enter a steady condition. (*Giorgi, Page 1244, Right column, Second paragraph*).

The second cited portion of *Giorgi* never recites that a program being analyzed is simulated multiple times and that "memory usage statistical data" is generated based on the multiple simulated executions of the program. Rather, this portion of *Giorgi* simply indicates

that some number of "memory references" occur "without an outcome." At most, this may indicate that several memory references to a cache memory in *Giorgi* can be simulated before usage data is collected. It in no way indicates that simulation of a program occurs multiple times.

For these reasons, *Giorgi* fails to anticipate the Appellant's invention as recited in Claim 5.

Claim 6

Claim 6 recites:

> The apparatus as set forth in Claim 1 wherein said memory optimization controller is further capable of determining at least one figure of merit associated with said at least one memory configuration, wherein said at least one figure of merit indicates a degree to which said at least one memory configuration satisfies said one or more design criteria.

Claim 6 depends from Claim 1 and is patentable for the reasons discussed above with respect to Claim 1.

For these reasons, *Giorgi* fails to anticipate the Appellant's invention as recited in Claim 6.

Claim 7

Claim 7 recites:

> The apparatus as set forth in Claim 1 further comprising a code optimization controller capable of modifying said program

> in response to said comparison of said memory usage statistical data and said one or more design criteria to thereby enable said embedded processing system to execute said program according to said one or more design criteria.

Claim 7 depends from Claim 1 and is patentable for the reasons discussed above with respect to Claim 1.

Moreover, the portion of *Giorgi* cited by the Examiner (page 1245, right column, paragraph two) refers to the selection of a "cache scheme" (including a mapping policy and a replacement algorithm). However, this portion of *Giorgi* is specifically referring to the selection of a cache scheme by a user. This action has nothing to do with a "code optimization controller" operating in the system of *Giorgi*.

Not only that, the selection of the "cache scheme" occurs so that it can be shown how a cache can exploit "program locality," which is determined using a "locality analysis" and the results of a simulation. (*Giorgi, Page 1244, Right column, Third paragraph – Page 1245, Right column, Second paragraph*). The "locality analysis" being performed (and the selection of the cache scheme based on it) has nothing to do with comparing "memory usage statistical data" and "one or more design criteria." Rather, the "locality analysis" is performed without any determinations that are based on comparing "memory usage statistical data" and "one or more design criteria."

In addition, *Giorgi* never recites that its system can modify a "cache scheme" selected by the user based on a comparison of "memory usage statistical data" and "one or more design criteria." At most, *Giorgi* appears to perform simulations using different "cache schemes" and

makes all simulation data available to the user. (*See, e.g., Giorgi, Page 1247, Left column, Third paragraph – Page 1248, Left column, First paragraph*). This is done without modify the "cache scheme" selected by the user based on a comparison of "memory usage statistical data" and "one or more design criteria."

For these reasons, *Giorgi* fails to anticipate the Appellant's invention as recited in Claim 7.

Claim 8

Claim 8 recites a method of designing a memory configuration for use in an embedded processing system, which includes:

> simulating execution of a program to be executed by the embedded processing system;
> monitoring, during the simulated execution of the program, memory accesses to a simulated memory space, wherein said memory accesses comprise read operations and write operations;
> generating memory usage statistical data associated with the monitored memory accesses;
> comparing the memory usage statistical data and one or more design criteria associated with the embedded processing system; and
> in response to the comparison, determining at least one memory configuration capable of satisfying the one or more design criteria.

Claim 8 contains similar recitations as Claim 1. As noted above with respect to Claim 1, the system of *Giorgi* is not capable of comparing "memory usage statistical data" and "one or more design criteria." The system of *Giorgi* is also not capable of determining, in response to the comparison, "at least one memory configuration" capable of satisfying the one or more

design criteria. At most, *Giorgi* simply performs simulations using multiple types of cache memory configurations and forces a user to view the simulation results. This fails to anticipate comparing "memory usage statistical data" and "one or more design criteria" associated with an embedded processing system and, "in response to the comparison," determining "at least one memory configuration capable of satisfying the one or more design criteria" as recited in Claim 8.

For these reasons, *Giorgi* fails to anticipate the Appellant's invention as recited in Claim 8 (and its dependent claims).

Claim 9

Claim 9 recites:

> The method as set forth in Claim 8 wherein the at least one memory configuration is determined from a set of memory types, the set of memory types comprising at least two of static random access memory (SRAM), dynamic random access memory (DRAM), read-only memory (ROM), flash RAM (FLASH), and electronically erasable programmable read-only memory (EEPROM).

Claim 9 depends from Claim 8 and is patentable for the reasons discussed above with respect to Claim 8.

Moreover, as noted above with respect to Claim 2, the portion of *Giorgi* cited by the Examiner (page 1246, right column, paragraph three) simply refers to various external memory modules in a system, such as a 1-MByte DRAM bank and a 120-KByte PROM bank. The system of *Giorgi* does not select these external memory modules "in response to" a comparison

of "memory usage statistical data" and "one or more design criteria." Instead, a user selects the external memory modules before any simulations take place.

For these reasons, *Giorgi* fails to anticipate the Appellant's invention as recited in Claim 9 (and its dependent claims).

Claim 10

Claim 10 recites:

> The method as set forth in Claim 9 wherein the at least one memory configuration comprises a first memory type and a first memory size associated with the first memory type.

Claim 10 depends from Claim 9 and is patentable for the reasons discussed above with respect to Claim 9.

For these reasons, *Giorgi* fails to anticipate the Appellant's invention as recited in Claim 10 (and its dependent claims).

Claim 11

Claim 11 recites:

> The method as set forth in Claim 10 wherein the at least one memory configuration further comprises a second memory type and a second memory size associated with the second memory type.

Claim 11 depends from Claim 10 and is patentable for the reasons discussed above with respect to Claim 10.

Moreover, as described above with respect to Claim 4, Claim 11 (through its dependence from Claim 10) recites that the "at least one memory configuration" includes a "first memory type" and an associated "first memory size" and a "second memory type" and an associated "second memory size." The portion of *Giorgi* cited by the Examiner (page 1246, right column, paragraph seven) simply refers to the user specifying various parameters of the external memory modules (such as the DRAM and PROM modules). These parameters are not based on any type of simulation or any analysis of simulation results, and they are selected before any simulations take place. (*See, e.g., Giorgi, Page 1246, Right column, Eighth paragraph*). These parameters therefore cannot possibly anticipate "at least one memory configuration" having multiple "memory types" and associated "sizes," where the "at least one memory configuration" is determined "in response to" a comparison of "memory usage statistical data" and "one or more design criteria."

For these reasons, *Giorgi* fails to anticipate the Appellant's invention as recited in Claim 11.

Claim 12

Claim 12 recites:

> The method as set forth in Claim 8 wherein the step of simulating execution of the program comprises the sub-steps of simulating execution of the program N times, wherein the step of monitoring the memory accesses comprises the sub-steps of monitoring the memory accesses during the N simulated executions of the program, and wherein the step of generating the

> memory usage statistical data is based on the N simulated executions of the program.

Claim 12 depends from Claim 8 and is patentable for the reasons discussed above with respect to Claim 8.

Moreover, as described above with respect to Claim 5, the portions of *Giorgi* cited by the Examiner (page 1244, left column, paragraph five and page 1244, right column, paragraph two) simply recite (i) the generation of a trace and (ii) the simulation of "an adequate number of memory references without an outcome." These portions of *Giorgi* never indicate that a program being analyzed is simulated multiple times and that "memory usage statistical data" is generated based on the multiple simulated executions of the program. At most, these portions of *Giorgi* may simply indicate that several memory references to a cache memory can be simulated before usage data is collected. It in no way indicates that simulation of a program occurs multiple times.

For these reasons, *Giorgi* fails to anticipate the Appellant's invention as recited in Claim 12.

Claim 13

Claim 13 recites:

> The method as set forth in Claim 8 further comprising the step of determining at least one figure of merit associated with the at least one memory configuration, wherein the at least one figure of merit indicates a degree to which the at least one memory configuration satisfies the one or more design criteria.

Claim 13 depends from Claim 8 and is patentable for the reasons discussed above with respect to Claim 8.

For these reasons, *Giorgi* fails to anticipate the Appellant's invention as recited in Claim 13.

Claim 14

Claim 14 recites:

> The method as set forth in Claim 8 further comprising the step of modifying the program in response to the comparison of the memory usage statistical data and the one or more design criteria to thereby enable the embedded processing system to execute the program according to the one or more design criteria.

Claim 14 depends from Claim 8 and is patentable for the reasons discussed above with respect to Claim 8.

Moreover, as noted above with respect to Claim 7, the portion of *Giorgi* cited by the Examiner (page 1245, right column, paragraph two) specifically refers to the selection of a "cache scheme" by a user so that it can be shown how a cache can exploit "program locality," which is determined using a "locality analysis" and the results of a simulation. (*Giorgi, Page 1244, Right column, Third paragraph – Page 1245, Right column, Second paragraph*). The "locality analysis" being performed (and the selection of the cache scheme based on it) has nothing to do with comparing "memory usage statistical data" and "one or more design criteria." Rather, the "locality analysis" is performed without any determinations that are based on comparing "memory usage statistical data" and "one or more design criteria."

Giorgi also never recites that its system can modify a "cache scheme" selected by the user based on a comparison of "memory usage statistical data" and "one or more design criteria." At most, *Giorgi* appears to perform simulations using different "cache schemes" and makes all simulation data available to the user. (*See, e.g., Giorgi, Page 1247, Left column, Third paragraph – Page 1248, Left column, First paragraph*). This is done without modify the "cache scheme" selected by the user based on a comparison of "memory usage statistical data" and "one or more design criteria."

For these reasons, *Giorgi* fails to anticipate the Appellant's invention as recited in Claim 14.

Claims 15-21

Claims 15-21 recite an "embedded processing system" designed according to the method as set forth in Claims 8-14, respectively. The Examiner bases the rejection of Claims 15-21 on the rejection of apparatus Claims 1-7. As shown above, Claims 1-7 and Claims 8-14 are patentable.

For these reasons, *Giorgi* fails to anticipate the Appellant's invention as recited in Claims 15-21.

Claim 22

Claim 22 recites a computer-readable storage medium containing computer-executable instructions for designing a memory configuration for use in an embedded processing system.

The computer-executable instructions include the steps of:

> simulating execution of a program to be executed by the embedded processing system;
>
> monitoring, during the simulated execution of the program, memory accesses to a simulated memory space, wherein said memory accesses comprise read operations and write operations;
>
> generating memory usage statistical data associated with the monitored memory accesses;
>
> comparing the memory usage statistical data and one or more design criteria associated with the embedded processing system; and
>
> in response to the comparison, determining at least one memory configuration capable of satisfying the one or more design criteria.

Claim 22 contains similar recitations as Claims 1 and 8. As noted above with respect to Claims 1 and 8, the system of *Giorgi* is not capable of comparing "memory usage statistical data" and "one or more design criteria." The system of *Giorgi* is also not capable of determining, in response to the comparison, "at least one memory configuration" capable of satisfying the one or more design criteria. At most, *Giorgi* simply performs simulations using multiple types of cache memory configurations and forces a user to view the simulation results. This fails to anticipate computer-executable instructions for "comparing the memory usage statistical data and one or more design criteria associated with the embedded processing system" and computer-executable instructions for "in response to the comparison, determining at least one memory configuration capable of satisfying the one or more design criteria" as recited in Claim 22.

For these reasons, *Giorgi* fails to anticipate the Appellant's invention as recited in Claim

22 (and its dependent claims).

Claim 23

Claim 23 recites:

> The computer-readable storage medium as set forth in Claim 22 wherein the at least one memory configuration is determined from a set of memory types, the set of memory types comprising at least two of static random access memory (SRAM), dynamic random access memory (DRAM), read-only memory (ROM), flash RAM (FLASH), and electronically erasable programmable read-only memory (EEPROM).

Claim 23 depends from Claim 22 and is patentable for the reasons discussed above with respect to Claim 22.

Moreover, as noted above with respect to Claims 2 and 9, the portion of *Giorgi* cited by the Examiner (page 1246, right column, paragraph three) simply refers to various external memory modules in a system, such as a 1-MByte DRAM bank and a 120-KByte PROM bank. The system of *Giorgi* does not select these external memory modules "in response to" a comparison of "memory usage statistical data" and "one or more design criteria." Instead, a user selects the external memory modules before any simulations take place.

For these reasons, *Giorgi* fails to anticipate the Appellant's invention as recited in Claim 23 (and its dependent claims).

Claim 24

Claim 24 recites:

> The computer-readable storage medium as set forth in Claim 23 wherein the at least one memory configuration comprises a first memory type and a first memory size associated with the first memory type.

Claim 24 depends from Claim 23 and is patentable for the reasons discussed above with respect to Claim 23.

For these reasons, *Giorgi* fails to anticipate the Appellant's invention as recited in Claim 24 (and its dependent claims).

Claim 25

Claim 25 recites:

> The computer-readable storage medium as set forth in Claim 24 wherein the at least one memory configuration further comprises a second memory type and a second memory size associated with the second memory type.

Claim 25 depends from Claim 24 and is patentable for the reasons discussed above with respect to Claim 24.

Moreover, as described above with respect to Claims 4 and 11, Claim 25 (through its dependence from Claim 24) recites that the "at least one memory configuration" includes a "first memory type" and an associated "first memory size" and a "second memory type" and an associated "second memory size." The portion of *Giorgi* cited by the Examiner (page 1246, right column, paragraph seven) simply refers to the user specifying various parameters of the external memory modules (such as the DRAM and PROM modules). These parameters are not based on any type of simulation or any analysis of simulation results, and they are selected

before any simulations take place. (*See, e.g., Giorgi, Page 1246, Right column, Eighth paragraph*). These parameters therefore cannot possibly anticipate "at least one memory configuration" having multiple "memory types" and associated "sizes," where the "at least one memory configuration" is determined "in response to" a comparison of "memory usage statistical data" and "one or more design criteria."

For these reasons, *Giorgi* fails to anticipate the Appellant's invention as recited in Claim 25.

Claim 26

Claim 26 recites:

> The computer-readable storage medium as set forth in Claim 22 wherein the step of simulating execution of the program comprises the sub-steps of simulating execution of the program N times, wherein the step of monitoring the memory accesses comprises the sub-steps of monitoring the memory accesses during the N simulated executions of the program, and wherein the step of generating the memory usage statistical data is based on the N simulated executions of the program.

Claim 26 depends from Claim 22 and is patentable for the reasons discussed above with respect to Claim 22.

Moreover, as described above with respect to Claims 5 and 12, the portions of *Giorgi* cited by the Examiner (page 1244, left column, paragraph five and page 1244, right column, paragraph two) simply recite (i) the generation of a trace and (ii) the simulation of "an adequate number of memory references without an outcome." These portions of *Giorgi* never indicate

that a program being analyzed is simulated multiple times and that "memory usage statistical data" is generated based on the multiple simulated executions of the program. At most, these portions of *Giorgi* may simply indicate that several memory references to a cache memory can be simulated before usage data is collected. It in no way indicates that simulation of a program occurs multiple times.

For these reasons, *Giorgi* fails to anticipate the Appellant's invention as recited in Claim 26.

Claim 27

Claim 27 recites:

> The computer-readable storage medium as set forth in Claim 22 further comprising the step of determining at least one figure of merit associated with the at least one memory configuration, wherein the at least one figure of merit indicates a degree to which the at least one memory configuration satisfies the one or more design criteria.

Claim 27 depends from Claim 22 and is patentable for the reasons discussed above with respect to Claim 22.

For these reasons, *Giorgi* fails to anticipate the Appellant's invention as recited in Claim 27.

Claim 28

Claim 28 recites:

> The computer-readable storage medium as set forth in Claim 22 further comprising the step of modifying the program in response to the comparison of the memory usage statistical data and the one or more design criteria to thereby enable the embedded processing system to execute the program according to the one or more design criteria.

Claim 28 depends from Claim 22 and is patentable for the reasons discussed above with respect to Claim 22.

Moreover, as noted above with respect to Claims 7 and 14, the portion of *Giorgi* cited by the Examiner (page 1245, right column, paragraph two) specifically refers to the selection of a "cache scheme" by a user so that it can be shown how a cache can exploit "program locality," which is determined using a "locality analysis" and the results of a simulation. (*Giorgi, Page 1244, Right column, Third paragraph – Page 1245, Right column, Second paragraph*). The "locality analysis" being performed (and the selection of the cache scheme based on it) has nothing to do with comparing "memory usage statistical data" and "one or more design criteria." Rather, the "locality analysis" is performed without any determinations that are based on comparing "memory usage statistical data" and "one or more design criteria."

For these reasons, *Giorgi* fails to anticipate the Appellant's invention as recited in Claim 28.

Claim 29

Claim 29 recites:

> The apparatus of Claim 1, wherein the memory usage statistical data comprises at least one of:

> one or more first histograms based on variable names contained in the program to be executed by the embedded processing system; and
>
> one or more second histograms based on memory locations accessed by the program to be executed by the embedded processing system.

Claim 29 depends from Claim 1 and is patentable for the reasons discussed above with respect to Claim 1.

Moreover, the portion of *Giorgi* cited by the Examiner (page 1245, Figure 2) simply represents the results of a "locality analysis." As noted above, *Giorgi* specifically refers to the selection of a "cache scheme" by a user for exploiting program locality. The "locality analysis" being performed here has nothing to do with comparing "memory usage statistical data" and "one or more design criteria." Rather, the "locality analysis" is performed without any determinations that are based on comparing "memory usage statistical data" and "one or more design criteria."

In addition, Figure 2 of *Giorgio* does not appear to be used during the simulations described on page 1247 of *Giorgio* (where different cache memory configurations are simulated). The contents of Figure 2 in *Giorgi* therefore cannot anticipate the "memory usage statistical data" recited in Claim 1. This is because the Examiner already asserted (with respect to Claim 1) that the "at least one memory configuration" relates to the cache memory configuration selected by the user on page 1247 of *Giorgi.* The Examiner must therefore show that the contents of Figure 2 in *Giorgi* are compared to "one or more design criteria" so as to determine "at least one memory configuration" during the simulations and other actions on page

1247 of *Giorgi*. The Examiner has not made this showing.

For these reasons, *Giorgi* fails to anticipate the Appellant's invention as recited in Claim 29.

Grouping of Claims

The claims on appeal do not stand or fall together, as may be seen from the arguments set forth above. Different claims have been argued separately under separate subheadings, and each set of claims should be considered separately. While the Appellant recognizes that a formal statement regarding the grouping of claims is no longer required, each claim should be considered separately, or at the very least each claim that is argued separately in the preceding sections of this brief should be considered separately. The fact that the claims have been argued separately shows that, if their patentability is not considered separately, any adverse decision would show that the limitations of some claims had been unfairly ignored.

REQUESTED RELIEF

The Board is respectfully requested to reverse the outstanding rejections and return this application to the Examiner for allowance.

The Commissioner is hereby authorized to charge any additional fees connected with this communication (including any extension of time fees) or credit any overpayment to Munck Butrus Deposit Account No. 50-0208.

Respectfully submitted,

MUNCK BUTRUS, P.C.

Date: Nov 13, 2006

William A. Munck
Registration No. 39,308
Attorney for Appellant

P.O. Drawer 800889
Dallas, Texas 75380
Phone: (972) 628-3600
Fax: (972) 628-3616
E-mail: *wmunck@munckbutrus.com*

DOCKET NO. 99-B-186 **PATENT**
CLIENT NO. STMI01-99186
Customer No. 23990

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of:	Vidyabhusan Gupta
Serial No.:	09/591,621
Filed:	June 9, 2000
For:	SYSTEM AND METHOD FOR DESIGNING AND OPTIMIZING THE MEMORY OF AN EMBEDDED PROCESSING SYSTEM
Technology Center:	2100
Group No.:	2128
Examiner:	Herng-der Day

APPENDIX A - Claims Appendix

1. An apparatus for designing a memory configuration for use in an embedded processing system comprising:

a simulation controller capable of simulating execution of a program to be executed by said embedded processing system;

a memory access monitor capable of monitoring, during said simulated execution of said program, memory accesses to a simulated memory space, wherein said memory access monitor

is capable of generating memory usage statistical data associated with said monitored memory accesses, and wherein said memory accesses comprise read operations and write operations; and

a memory optimization controller capable of comparing said memory usage statistical data and one or more design criteria associated with said embedded processing system and, in response to said comparison, determining at least one memory configuration capable of satisfying said one or more design criteria.

2. The apparatus as set forth in Claim 1 wherein said at least one memory configuration is determined from a set of memory types, said set of memory types comprising at least two of static random access memory (SRAM), dynamic random access memory (DRAM), read-only memory (ROM), flash RAM (FLASH), and electronically erasable programmable read-only memory (EEPROM).

3. The apparatus as set forth in Claim 2 wherein said at least one memory configuration comprises a first memory type and a first memory size associated with said first memory type.

4. The apparatus as set forth in Claim 3 wherein said at least one memory configuration further comprises a second memory type and a second memory size associated with said second memory type.

5. The apparatus as set forth in Claim 1 wherein said simulation controller simulates execution of said program N times and wherein said memory access monitor monitors said memory accesses during said N simulated executions of said program and generates said memory usage statistical data based on said N simulated executions of said program.

6. The apparatus as set forth in Claim 1 wherein said memory optimization controller is further capable of determining at least one figure of merit associated with said at least one memory configuration, wherein said at least one figure of merit indicates a degree to which said at least one memory configuration satisfies said one or more design criteria.

7. The apparatus as set forth in Claim 1 further comprising a code optimization controller capable of modifying said program in response to said comparison of said memory usage statistical data and said one or more design criteria to thereby enable said embedded processing system to execute said program according to said one or more design criteria.

8. A method of designing a memory configuration for use in an embedded processing system, the method comprising the steps of:

simulating execution of a program to be executed by the embedded processing system;

monitoring, during the simulated execution of the program, memory accesses to a simulated memory space, wherein said memory accesses comprise read operations and write operations;

generating memory usage statistical data associated with the monitored memory accesses;

comparing the memory usage statistical data and one or more design criteria associated with the embedded processing system; and

in response to the comparison, determining at least one memory configuration capable of satisfying the one or more design criteria.

9. The method as set forth in Claim 8 wherein the at least one memory configuration is determined from a set of memory types, the set of memory types comprising at least two of static random access memory (SRAM), dynamic random access memory (DRAM), read-only memory (ROM), flash RAM (FLASH), and electronically erasable programmable read-only memory (EEPROM).

10. The method as set forth in Claim 9 wherein the at least one memory configuration comprises a first memory type and a first memory size associated with the first memory type.

11. The method as set forth in Claim 10 wherein the at least one memory configuration further comprises a second memory type and a second memory size associated with the second memory type.

12. The method as set forth in Claim 8 wherein the step of simulating execution of the program comprises the sub-steps of simulating execution of the program N times, wherein the step of monitoring the memory accesses comprises the sub-steps of monitoring the memory accesses during the N simulated executions of the program, and wherein the step of generating the memory usage statistical data is based on the N simulated executions of the program.

13. The method as set forth in Claim 8 further comprising the step of determining at least one figure of merit associated with the at least one memory configuration, wherein the at least one figure of merit indicates a degree to which the at least one memory configuration satisfies the one or more design criteria.

14. The method as set forth in Claim 8 further comprising the step of modifying the program in response to the comparison of the memory usage statistical data and the one or more design criteria to thereby enable the embedded processing system to execute the program according to the one or more design criteria.

15. An embedded processing system designed according to the method as set forth in Claim 8.

16. An embedded processing system designed according to the method as set forth in Claim 9.

17. An embedded processing system designed according to the method as set forth in Claim 10.

18. An embedded processing system designed according to the method as set forth in Claim 11.

19. An embedded processing system designed according to the method as set forth in Claim 12.

20. An embedded processing system designed according to the method as set forth in Claim 13.

21. An embedded processing system designed according to the method as set forth in Claim 14.

22. For use in a processing system, a computer-readable storage medium containing computer-executable instructions for designing a memory configuration for use in an embedded processing system, the computer-executable instructions comprising the steps of:

simulating execution of a program to be executed by the embedded processing system;

monitoring, during the simulated execution of the program, memory accesses to a simulated memory space, wherein said memory accesses comprise read operations and write operations;

generating memory usage statistical data associated with the monitored memory accesses;

comparing the memory usage statistical data and one or more design criteria associated with the embedded processing system; and

in response to the comparison, determining at least one memory configuration capable of satisfying the one or more design criteria.

23. The computer-readable storage medium as set forth in Claim 22 wherein the at least one memory configuration is determined from a set of memory types, the set of memory types comprising at least two of static random access memory (SRAM), dynamic random access memory (DRAM), read-only memory (ROM), flash RAM (FLASH), and electronically erasable programmable read-only memory (EEPROM).

24. The computer-readable storage medium as set forth in Claim 23 wherein the at least one memory configuration comprises a first memory type and a first memory size associated with the first memory type.

25. The computer-readable storage medium as set forth in Claim 24 wherein the at least one memory configuration further comprises a second memory type and a second memory size associated with the second memory type.

26. The computer-readable storage medium as set forth in Claim 22 wherein the step of simulating execution of the program comprises the sub-steps of simulating execution of the program N times, wherein the step of monitoring the memory accesses comprises the sub-steps of monitoring the memory accesses during the N simulated executions of the program, and wherein the step of generating the memory usage statistical data is based on the N simulated executions of the program.

27. The computer-readable storage medium as set forth in Claim 22 further comprising the step of determining at least one figure of merit associated with the at least one memory configuration, wherein the at least one figure of merit indicates a degree to which the at least one memory configuration satisfies the one or more design criteria.

28. The computer-readable storage medium as set forth in Claim 22 further comprising the step of modifying the program in response to the comparison of the memory usage statistical data and the one or more design criteria to thereby enable the embedded processing system to execute the program according to the one or more design criteria.

29. The apparatus of Claim 1, wherein the memory usage statistical data comprises at least one of:

one or more first histograms based on variable names contained in the program to be executed by the embedded processing system; and

one or more second histograms based on memory locations accessed by the program to be executed by the embedded processing system.

DOCKET NO. 99-B-186 **PATENT**
CLIENT NO. STMI01-99186
Customer No. 23990

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of:	Vidyabhusan Gupta
Serial No.:	09/591,621
Filed:	June 9, 2000
For:	SYSTEM AND METHOD FOR DESIGNING AND OPTIMIZING THE MEMORY OF AN EMBEDDED PROCESSING SYSTEM
Technology Center:	2100
Group No.:	2128
Examiner:	Herng-der Day

APPENDIX B
Evidence Appendix

none

DOCKET NO. 99-B-186
CLIENT NO. STMI01-99186
Customer No. 23990

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of:	Vidyabhusan Gupta
Serial No.:	09/591,621
Filed:	June 9, 2000
For:	SYSTEM AND METHOD FOR DESIGNING AND OPTIMIZING THE MEMORY OF AN EMBEDDED PROCESSING SYSTEM
Technology Center:	2100
Group No.:	2128
Examiner:	Herng-der Day

APPENDIX C
Related Proceedings Appendix

none